https://doi.org/10.7840/kics.2023.48.6.712

모델 프리 강화학습 정책을 적용한 호기심 기반 TD-MPC

지 창 훈•, 김 주 봉*, 한 연 희°

Curiosity-Driven TD-MPC with Model-Free Reinfocement Learning Policy

Chang-Hun Ji•, Ju-Bong Kim*, Youn-Hee Han°

요 약

최근 모델 기반 강화학습 알고리즘 중 가장 높은 성능을 가지고 있는 TD-MPC는 학습 과정에서 모델 예측 제어와 DDPG 에이전트로부터 행동을 추출한다. 하지만 DDPG 에이전트는 추출된 행동은 환경에 적용되지 않고, 모델 예측 제어로부터 추출된 행동만 환경에 적용한다. 본 논문에서는 TD-MPC가 가지고 있는 DDPG 에이전트와 모델 예측 제어를 모두 고려하여 환경에 적용하는 이중 정책을 활용한 향상된 TD-MPC를 제안한다. 또한, 호기심 기반으로 탐험을 장려하여 이중 정책 사이에서 행동을 선택할 때 발생할 수 있는 활용의 편향을 해결하였다. DeepMind Control Suite의 여러 환경에서 제안하는 알고리즘이 기존의 TD-MPC보다 높은 샘플 효율성과 높은 성능을 가지고 있음을 확인한다.

Key Words : Deep Reinforcement Learning, Model-based Reinforcement Learning, Model Predictive Control, Precise Control

ABSTRACT

TD-MPC, which has the highest performance among recent model-based reinforcement learning algorithms, extracts behaviors from model predictive control and DDPG agents in the learning process. However, the DDPG agent does not apply the extracted behavior to the environment, but only applies the behavior extracted from model predicted control to the environment. In this paper, we propose an enhanced TD-MPC that utilizes a dual policy that applies to the environment by considering both the DDPG agent and model predictive control of TD-MPC. In addition, by encouraging exploration based on curiosity, bias in utilization that can occur when choosing an action between dual policies is addressed. It is confirmed that the algorithm proposed in various environments of the DeepMind Control Suite has higher sample efficiency and higher performance than the existing TD-MPC.

※ 이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2018R1A6A1A03025526 & NRF-2023R1A2C1003143).

◆ First Author : Future Convergence Engineering, Korea University of Technology and Education, koir5660@koreatech.ac.kr, 학생회원

° Corresponding Author: Future Convergence Engineering, Korea University of Technology and Education, yhhan@koreatech.ac.kr, 종신회원

* Future Convergence Engineering, Korea University of Technology and Education, rlawnqhd@koreatech.ac.kr, 학생회원

논문번호 : 202212-303-C-RN, Received December 16, 2022; Revised February 6, 2023; Accepted February 19, 2023

Ⅰ. 서 론

강화학습은 금융부터 로봇 제어까지 많은 분야에서 활용되고 있다. 그중 로봇 제어는 전통적인 기계 분야와 환경 간의 상호작용을 포함하기 때문에 강화학습으로 학습이 이뤄지기 어려운 분야이다. 로봇 제어 분야에서 강화학습 중에 이뤄지는 환경 간의 상호작용은 로봇에게 과도한 부하를 줄 수가 있고, 불필요한 에너지 소비를 하게 만든다. 하지만 강화학습은 로봇 시스템의 적응성(Adaptability)을 높여 로봇 시스템의 복잡한 환경을 효과적으로 제어할 수 있는 인공지능 프레임워크이다[1]. 이러한 장점으로 강화학습은 실제 환경 로봇 시스템의 적용을 목적으로 물리적인 시스템에서 많은 연구가 이루어지고 있다[2-5].

앞서 언급한 강화학습의 단점은 환경과의 상호작용을 줄이면서 학습을 진행함으로 최소화시킬 수 있다. 학습 성공까지 필요한 샘플(Sample)의 비율을 샘플 효율성(Sample Efficiency)이라 하는데, 샘플 효율성이 높으면 강화학습의 단점을 최소화시킬 수 있다. 환경과 행동을 결정하는 에이전트와의 끊임없는 상호작용을 통해 학습을 진행하는 강화학습에서 샘플 효율성을 높이는 것은 그동안 해결해야 할 과제였다. 최근 환경 모델과 계획(Planning)을 활용하여 학습을 진행하는 모델 기반 강화학습은 샘플 효율성이 높다는 장점이 있어 로봇 제어와 관련하여 많은 연구가 이루어졌다. 최근 모델 기반 강화학습에서 가장 좋은 성능을 보이는 알고리즘 중 하나로 모델 예측 제어를 위한 시간차 학습(Temperal Difference Learning for Model Predictive Control, TD-MPC)[6]이 있다.

TD-MPC는 행동을 추출하는 정책을 2개 가지고 있다. 하나는 모델 예측 제어(Model Predictive Control, MPC)[7]를 활용하는 모델 기반 정책이고, 다른 하나는 심층 결정론적 정책경사법(Deep Deterministic Policy Gradient, DDPG)[8]이적용된 모델 프리 정책이다. 모델 기반 강화학습에 해당하는 MPC의 행동과 모델 프리 강화학습에 해당하는 DDPG의 행동은 서로 다른 특징을 가지고 있다. 하지만 TD-MPC는 모델 기반 강화학습의 장점을 적용하기 위해 오직 MPC로부터 추출된 행동만을 환경에 적용한다.

따라서 본 논문은 TD-MPC가 가지고 있는 두 개의 정책으로부터 추출된 행동을 모두 환경에 적용하는 새로운 알고리즘을 제안한다. 또한, 이때 일어날 수 있는 탐험-활용 균형(Exploration-Exploitation Balance)이 활용으로 지나치게 편향되는 것을 방지하고자 새로운 탐험 기법인 호기심 기반(Curiosity-driven) 탐험 기법을 TD-MPC에 적용한다. 이 모든 과정은 TD-MPC에 추가적인 심층 신경망 없이 수행된다.

새롭게 제안되는 알고리즘은 가상 물리 시뮬레이션 환경인 DeepMind Control Suite[2]의 다양한 환경에서 실험이 진행된다. DeepMind Control Suite는 강화학습에서 가상 물리 시뮬레이션 환경을 제공하는 대표적인 벤치마크이다. 실험 결과는 TD-MPC보다 제안하는 알고리즘이 샘플 효율성과 성능 면에서 월등하다는 것을 보여준다.

Ⅱ. 관련 연구

2.1 모델 기반 강화학습

강화학습의 문제는 마르코프 의사결정 과정(Markov Decision Process, MDP) 최적화 문제로 정의될 수 있다[9]. 마르코프 의사결정 과정의 역학(Dynamics) 모델을 안다면, 강화학습 에이전트는 현재 시점으로부터 가상으로 원하는 행동의 트레젝토리(Trajectory of Actions)를 계획 할 수 있다. 마르코프 의사결정 과정의 역학 모델은 환경 모델이라고도 한다. 환경 모델과 계획을 활용하는 강화학습을 모델 기반(Model-based) 강화학습이라고 정의한다[10].

모델 기반 강화학습은 계획을 통해 샘플 효율성을 높일 수 있다. 또한, 탐험과 성능 측면에도 장점이 있다[11]. 이러한 장점을 이유로 모델 기반 강화학습은 순차적 의사결정 문제의 해결에 좋은 효과를 보였다. 대표적으로 게임이나, 연속적인 제어 환경에서 기존 모델 기반 강화학습 연구들은 계획을 사용하여 강화학습 에이전트를 효과적으로 학습시키는 데 성공했다[12-15]. 환경 모델은 보통 환경으로부터 주어지지 않고 학습을 통해 얻는다. 모델 기반 강화학습은 환경 모델의 정확도에 크게 의존한다는 단점이 있다. 따라서 환경 모델이 정확하게 학습되지 않는다면, 원하는 성능을 달성하기 힘들다.

모델 기반 강화학습은 정확한 환경 모델의 학습을 위해 제어 알고리즘을 비롯한 다양한 최적화 알고리즘을 계획에 적용하기도 한다. MuZero[16]와 Efficient Zero[17]에서의 몬테카를로 트리 탐색(Monte Carlo Tree Search)[18]과 TD-MPC에서 MPC가 이에 해당한다.

2.2 호기심 기반 강화학습

학습 도중 실제 강화학습 환경에서 에이전트가 환다. 이때, 호기심(Curiosity)은 에이전트가 환경을 탐색하고 나중에 학습에 유용한 경험을 하게 유도하는

내적 보상(Intrinsic Reward)으로 작용할 수 있다[19]. 현재 연구된 호기심 대부분은 현재 상태에 대해 특정 행동을 할 때, 다음 상태를 잘 예측할 수 있는지를 측정한다[19-21]. 다음 상태를 잘 예측할 수 있으면, 해당 경험에 대해 익숙하다고 판단해서 내적 보상을 적게 주고, 다음 상태를 잘 예측할 수 없으면, 해당 경험에 대해 탐험이 필요하다고 생각해 내적 보상을 크게 준다.

내적 호기심 모듈(Intrinsic Curiosity Module, ICM)은 호기심 기반 강화학습의 일종이다. ICM은 역동역학(Inverse Dynamics Model) 모델과 역학(Dynamics) 모델 그리고 특징(Feature) 모델로부터 내적 보상을 추출한다[19]. 역학 모델은 현재 상태와 행동이 들어오면 다음 상태를 예측한다. 특징 모델은 상태에서 내적 보상을 구하는데 필요한 특징들을 추출하고, 마지막 역 동역학 모델은 역학 모델과 특징 모델에게 현재 상태로부터 다음 상태와 행동의 연관성을 알려주는 역할을 한다. 본 논문에서 제안하는 알고리즘은 ICM을 활용하여 호기심 기반 강화학습을 구현하였다.

2.3 모델 예측 제어를 위한 시간차 학습

TD-MPC는 모델 기반 강화학습 중 높은 성능을 가지고 있는 알고리즘 중 하나이다. TD-MPC는 환경 모델을 학습하고, 학습된 환경 모델을 활용하여 계획을 통해 환경에 적용될 행동을 추출한다. TD-MPC는 행동 가치 모델과 함께 환경 모델을 공동으로 학습하는 Task-Oriented Latent Dynamics (TOLD) 모델을 제안한다. TOLD 모델의 구성은 아래와 같다.

$$Representation:\ z_t = h_\theta(s_t) \tag{1}$$

$$Latent\ Dynamics:\ z_{t+1} = d_\theta(z_t, a_t) \tag{2}$$

$$Reward:\ \hat{r}_t = R_\theta(z_t, a_t) \tag{3}$$

$$Action\ Value:\ \hat{q}_t = Q_\theta(z_t, a_t) \tag{4}$$

$$Policy:\ \hat{a}_t = \pi_\theta(z_t) \tag{5}$$

TD-MPC는 MPC가 근시안적인 해결책만을 구할 수 있다는 단점을 극복하기 위해 행동 가치 모델을 통한 누적 보상의 기댓값을 고려하여 행동을 추출한다. TD-MPC는 MPC의 알고리즘으로 모델 예측 경로 적분(Model Predictive Path Integral, MPPI)[22] 알고리즘을 사용한다.

표 1. TD-MPC의 모델 기반 정책과 모델 프리 정책
Table 1. Model-based policy and Model-free policy in TD-MPC

	Model-based Policy(Π)	Model-free Policy(π)
Reinforcement Learning to Use	Model-based Reinforcement Learning	Model-free Reinforcement Learning
Action Extraction Method	Cross Entropy Method	Function Approximation
Utilized in TD-MPC	Extract Actions to Apply to the Environment	Sample Addition of MPC, Action Extraction for Action Value Function Training

TD-MPC는 MPC를 활용하는 모델 기반 정책과 DDPG를 활용하는 모델 프리 정책을 모두 가지고 있다. 두 개의 정책 모두 환경에 적용할 행동을 뽑을 수 있지만, TD-MPC는 환경에 적용할 행동을 추출하는데 모델 기반 정책만을 활용하고 있다. 표 1은 TD-MPC에 존재하는 모델 기반 정책과 모델 프리 정책을 비교한다.

Ⅲ. 제안하는 알고리즘

본 논문에서는 TD-MPC에 이중 정책의 사용과 호기심 기반 탐험 방법을 적용한 새로운 알고리즘을 제안한다. 아래에서 제안하는 알고리즘을 자세하게 설명한다. 최종적으로 제안하는 알고리즘의 개요도는 그림 1에서 제시한다. 또한, 그림 3에서는 제안하는 알고리즘의 의사코드를 보여준다.

3.1 이중 정책을 통한 행동 선택

3.1.1 이중 정책 TD-MPC

식 (1) - 식 (5)에서 알 수 있듯, TD-MPC를 구성하는 주요 모델들은 표현 모델, 환경 모델에 해당하는 잠재 역학 모델, 보상 모델, 행동 가치 모델 그리고 정책 모델이다. TD-MPC는 구성되는 모델을 활용하여 MPC를 통해 행동을 추출한다. TD-MPC는 행동을 추출할 수 있는 정책으로 모델 기반 정책과 모델 프리 정책을 가지고 있다. 하지만, 표 1에 표현되어 있듯이, 모델 기반 정책을 통해 결정된 행동만 환경에 적용된다. TD-MPC에서 모델 프리 정책으로 결정된 행동은

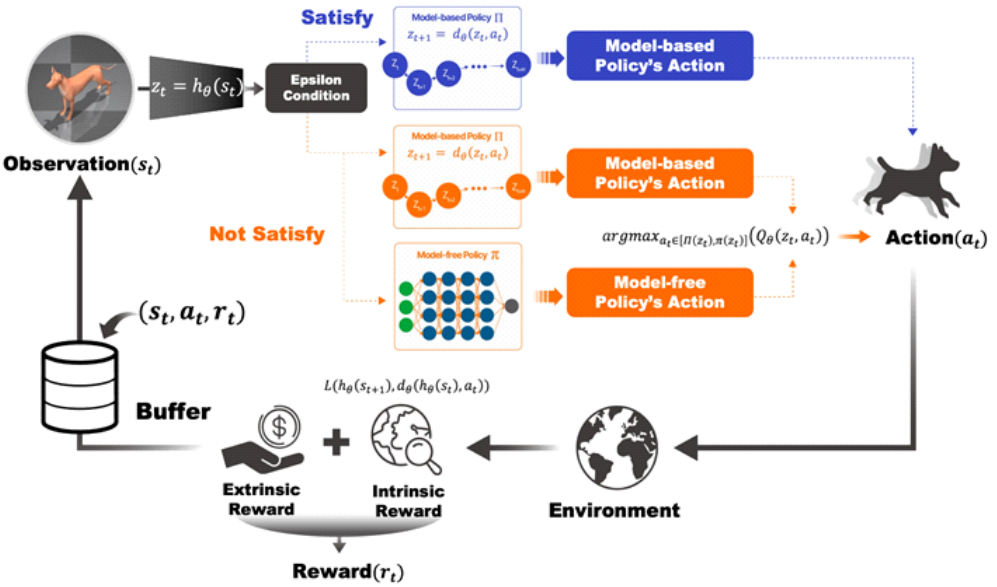


그림 1. 제안하는 알고리즘 개요도
Fig. 1. Schematic of proposed algorithm

환경에 적용되지 않는다. TD-MPC에서 모델 프리 정책은 MPC의 샘플을 추가하고, 행동 가치 모델을 학습시키는 제한적인 부분에 활용된다.

TD-MPC는 모델 프리 정책으로 DDPG 알고리즘을 활용한다. 이렇게 학습된 모델 프리 정책은 충분히 환경에 적용될 수 있는 행동을 추출할 수 있다. 또한, 모델 기반 정책보다 안 좋다고 판단될 수 없다. 실제로 많은 강화학습 분야에서 간단한 구조의 모델 프리 정책은 모델 기반 정책보다 높은 성능을 나타내고 있다.

따라서 본 논문에서는 TD-MPC의 2개의 정책을 모두 활용하는 이중 정책 활용을 제안한다. TD-MPC는 누적 보상의 개념에서 행동의 가치를 판단할 수 있는 행동 가치 모델을 학습시키고 있다. 제안하는 이중 정책 TD-MPC는 모델 프리 정책의 행동과 모델 기반 정책의 행동 중 더 가치 있는 행동을 환경에 적용되는 행동으로 정의하는 새로운 행동 추출 방법을 제시한다.

$$a_t = argmax_{\ a_t \in [\Pi(z_t), \pi(z_t)}(Q_\theta(z_t, a_t)) \tag{6}$$

이중 정책 TD-MPC는 기존 TD-MPC와 비교하여 추가적인 모델 복잡도와 계산복잡도 없이 모델 프리 정책 π의 행동과 모델 기반 정책 Π의 행동 중 더 가치 있는 행동을 선택할 수 있다.

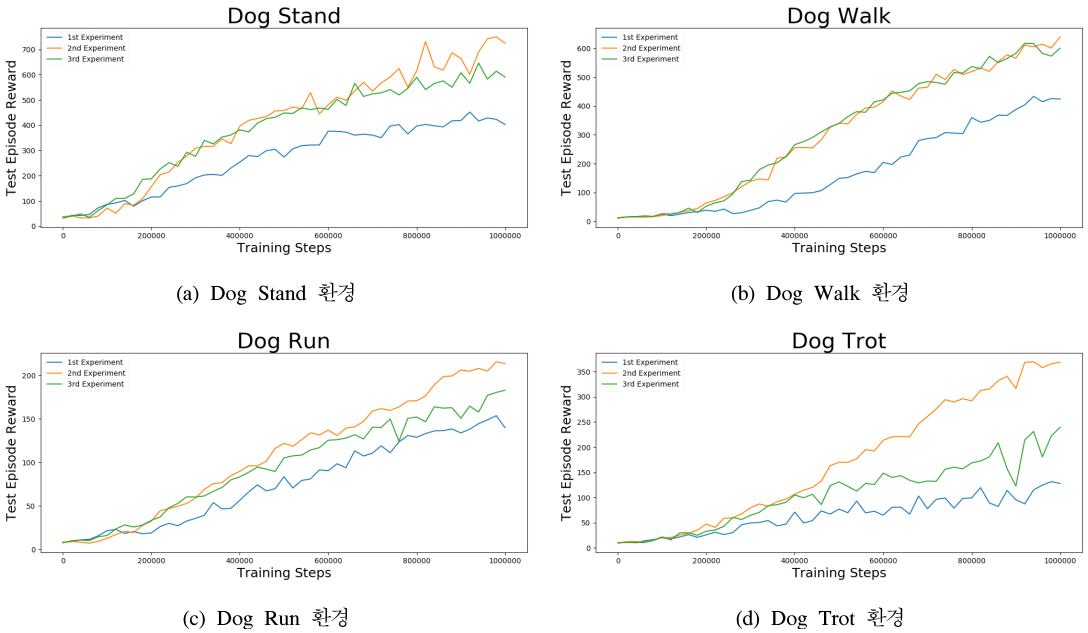
3.1.2 이중 정책 TD-MPC의 행동 선택 비율 조정

모델 기반 강화학습의 가장 큰 장점 중 하나는 샘플 효율성이 좋다는 것이다[11]. 샘플 효율성이 좋다는 말은 환경과 에이전트의 적은 상호작용으로 학습을 빠르게 진행하게 한다는 의미와 같다. 따라서 모델 기반 정책의 학습이 잘 수행된다면, 학습 초반에 모델 프리 정책으로 추출된 샘플보다 모델 기반 정책으로 추출된 샘플이 더 가치 있는 샘플일거라고 예측할 수 있다. 그리고 TD-MPC에서 모델 기반 정책으로부터 나온 가치 있는 샘플들은 모델 프리 정책의 학습도 더욱 가속 시킬 수 있다고 예측할 수 있다.

이를 증명하기 위해, DeepMind Control Suite 환경 중 상태 공간(Observation Shape)이 가장 높은 Dog 도메인의 4가지 환경에서 실험을 진행하였다. 실험은 총 3가지 방식으로 진행하였다. 3가지의 실험 방식은 다음과 같다.

① 학습 초반 모델 프리 정책의 행동 비율이 높음
② 학습 초반 모델 기반 정책의 행동 비율이 높음
③ 모델 기반 정책과 모델 프리 정책 같은 비율

실험들은 1,000,000 훈련 스텝(Training Step)이 되면 종료된다. 각 실험은 5번씩 진행되어 그림 2에서 평균으로 나타내었다. 첫 번째 실험과 두 번째 실험은 훈련이 진행되면서 정책 선택의 비율이 선형적으로 변경된다.



(a) Dog Stand 환경

(b) Dog Walk 환경

(c) Dog Run 환경

(d) Dog Trot 환경

그림 2. 행동 타입 비교 실험 결과 그래프
Fig. 2. The result graph comparing action type

결과 그래프는 각 실험에서 모은 샘플들을 활용하여 학습한 모델 프리 정책의 성능을 나타낸다. 세 가지의 실험을 진행하기 위해 학습 횟수에 따라 선형적으로 줄어드는 ϵ을 활용하였다. 학습 횟수에 따른 ϵ의 정의는 다음과 같다.

$$\epsilon = (1.0 - \mathrm{mix}) \times 1.0 + \mathrm{mix} \times 0.5 \tag{7}$$

식 (7)에서 mix는 ϵ이 1.0에서 0.5까지 훈련 스텝에 선형적으로 감소하도록 한다. t가 훈련 스텝일 때 mix는 다음 식 (8)로 결정된다.

$$\mathrm{mix} = \mathrm{Clip}\left(\frac{\mathrm{t}}{1{,}000{,}000}, 0.5, 1.0\right) \tag{8}$$

위 식 (7)과 (8)에 따르면 ϵ은 1.0에서 시작하여 0.5까지 1,000,000 훈련 스텝 동안 훈련 스텝에 따라 선형적으로 감소한다. 식 (9) - 식 (11)은 차례대로 첫 번째 실험과 두 번째 실험, 세 번째 실험의 정책 선택 방법을 정의한다.

$$action = \begin{cases} \pi(z_t) & for\ random(0,1) < \epsilon \\ \Pi(z_t) & for\ ramdom(0,1) \geq \epsilon \end{cases} \tag{9}$$

$$action = \begin{cases} \Pi(z_t) & for\ random(0,1) < \epsilon \\ \pi(z_t) & for\ ramdom(0,1) \geq \epsilon \end{cases} \tag{10}$$

$$action = \begin{cases} \pi(z_t) & for\ random(0,1) < 0.5 \\ \Pi(z_t) & for\ ramdom(0,1) \geq 0.5 \end{cases} \tag{11}$$

첫 번째 실험은 학습 초반에 모델 프리 정책의 선택이 높은 비율로 선택되며, 두 번째 실험은 학습 초반에 모델 기반 정책의 선택이 높은 비율로 선택된다. 첫 번째 실험과 두 번째 실험은 ϵ에 의해 행동 선택의 비율이 선형적으로 변화한다. 그리고 마지막으로 세 번째 실험은 모델 기반 정책과 모델 프리 정책의 선택이 균일하게 선택되었다.

그림 2는 각각 DeepMind Control Suite의 Dog-Walk, Dog-Stand, Dog-Run, Dog-Trot 환경의 실험 결과 그래프이다. 결과 그래프를 보면 학습 초반에 모델 기반 정책으로부터 행동을 추출하여 만들어진 샘플로 모델 프리 정책을 학습시킬 때 모델 프리 정책의 성능이 가장 높은 것을 확인할 수 있다. 또한, 만약 학습 초반에 모델 프리 정책의 행동이 모델 기반 정책의 행동보다 더 좋은 행동일지라도, 학습 초반 행동 가치 모델은 학습이 진행되지 않았기 때문에 어떤 행동이 더 좋은지 올바르게 구분하지 못한다.

따라서 본 논문은 학습 초반에는 행동 가치와 상관없이 모델 기반 정책의 행동을 환경에 적용하고 학습

이 진행되면서 모델 기반 정책과 모델 프리 정책의 행동 중 행동 가치 모델의 결괏값이 높은 행동을 환경에 적용하는 새로운 알고리즘을 제안한다. 구체적으로, 제안하는 알고리즘은 임의로 정한 초반 학습까지는 모델 기반 정책의 행동으로 학습을 시작한다. 그 이후에, ϵ을 0과 1 사이의 임의 실수와 비교하여 모델 기반 정책의 행동을 환경에 적용할 것인지, 이중 정책을 활용하여 행동 가치가 높은 행동을 환경에 적용할 것인지 결정된다. 학습 시간이 지날수록, 이중 정책을 활용하여 행동을 정하는 비율이 선형적으로 높아지게 된다. 추가로, 모델 기반 정책의 행동이 일정 이상 뽑히지 않을 경우, TD-MPC가 모델 기반 강화학습으로부터 얻을 수 있는 샘플 효율성, 탐험, 최적성과 같은 이점을 갖지 못하기 때문에, 일정 비율 이상은 항상 모델 기반 정책의 행동을 뽑도록 $\min\epsilon$을 설정하였다.

제안하는 알고리즘에서, 행동 선택 방법의 비율은 선형적으로 감소하는 ϵ을 통해 조절한다. 본 논문에서 최종적으로 사용하는 ϵ은 식 (12)와 같다.

$$\epsilon = (1.0 - \mathrm{mix}) \times 1.0 + \mathrm{mix} \times \min\epsilon \tag{12}$$

식 (12)에서 mix는 ϵ이 1.0에서 $\min\epsilon$까지 훈련 스텝 t에 선형적인 감소가 되도록 하고, 다음 식 (13)으로 결정된다.

$$\mathrm{mix} = \mathrm{Clip}\left(\frac{\mathrm{t}}{\mathrm{period}}, \min\epsilon, 1.0\right) \tag{13}$$

식 (13)에서 $\min\epsilon$과 period은 하이퍼 파라미터로 각각 ϵ의 최솟값과 ϵ이 감소하는 기간을 말한다. 즉 period 이후 학습부터는 ϵ은 $\min\epsilon$으로 유지된다. 최종적으로 환경에 적용되는 행동 $\ddot{a}_t$는 0과 1 사이의 임의의 수 x 에 대하여 다음과 같은 식 (14)에 의하여 결정된다.

$$\ddot{a}_t = \begin{cases} \Pi(z_t) & for\ x < \epsilon \\ argmax_{a_t \in [\Pi(z_t), \pi(z_t)]} (Q_\theta(z_t, a_t)) & for\ x \geq \epsilon \end{cases} \tag{14}$$

식 (14)는 그림 1 제안하는 알고리즘 개요도에서 Epsilon 조건을 통한 행동 선택으로 제시되었다. Epsilon 조건이 만족할 경우 강화학습 에이전트는 모델 프리 정책을 고려하지 않고 모델 기반 정책의 행동만 환경에 적용한다. Epsilon 조건을 만족하지 못하면 이중 정책과 행동 가치 모델을 활용하여 환경에 적용할 행동을 선택한다.

3.2 호기심 기반 탐험 장려

강화학습에서 탐험과 활용은 강화학습의 성능 향상을 위해 반드시 수행되어야 한다. 탐험은 강화학습 에이전트가 다양한 경험을 쌓기 위해 새로운 시도를 하는 것을 말하고, 활용은 현재까지의 경험 중 가장 최대의 보상을 얻을 수 있는 행동을 선택하는 것을 말한다. 탐험과 활용 둘 중 하나에 너무 집중하면, 균형이 무너져서 학습이 진행되지 않는다. 이를 강화학습의 탐험과 활용 딜레마라고 한다[10].

이중 정책 활용은 모델 기반 정책과 모델 프리 정책의 행동 중 행동 가치가 더 높은 행동을 탐욕적으로 선택한다. 이러한 행동 선택 방법은 활용을 극대화하며, 탐험과 활용의 균형을 깨뜨릴 수 있다. 구체적으로 기존 TD-MPC를 활용한 샘플들의 행동 가치 추정값은 이중 정책 TD-MPC를 사용한 샘플들의 행동 가치 추정값보다 항상 낮거나 같다. 따라서, 임의의 모든 (z_t, a_t)에 대하여 항상 다음 부등식이 성립한다.

$$Q_\theta(z_t, \Pi(z_t)) \leq max_{a_t \in [\Pi(z_t), \pi(z_t)]} (Q_\theta(z_t, a_t)) \tag{15}$$

이번 절에서는 식 (15)에 따른 활용의 편향을 막기 위해 새로운 탐험 방법을 제안한다.

호기심 기반 강화학습은 강화학습에 탐험을 장려하기 위해 널리 사용되는 방법이다. 호기심 기반 강화학습은 호기심 기반의 내적 보상을 만드는 새로운 모델을 추가한다. 본 논문은 새로운 모델을 추가하지 않고, TD-MPC 기존의 모델들을 활용하여 호기심 기반의 새로운 내적 보상을 만드는 알고리즘을 제안한다.

본 논문에서, TD-MPC에 적용한 호기심 기반 알고리즘은 해당 호기심 기반 알고리즘은 제2장에서 설명한 ICM을 사용하였다. ICM은 잠재 역학 모델을 이용하여 현재 상태와 환경에 적용될 행동으로 다음 상태를 예측한다. 예측된 다음 상태를 실재 다음 상태와 비교하여 둘의 평균 제곱 오차를 내적 보상으로 정의한다.

$$reward_{intrinsic} = (h_\theta(s_{t+1}) - d_\theta(s_t, a_t))^2 \tag{16}$$

환경에서 받는 보상을 외적 보상(Extrinsic Reward)이라고 한다면, 에이전트의 모델들이 최종적으로 학습하게 되는 보상 $reward_{total}$은 내적 보상과 외적 보상을 합한 보상이다.

$$reward_{total} = reward_{extrinsic} + reward_{intrinsic} \quad (17)$$

그림 1에서 버퍼에 저장되는 최종 보상이 외적 보상과 내적 보상을 합한 값이 되는 것으로 호기심 기반 탐험 장려를 표현하였다.

3.3 모델 구성 및 업데이트

제안하는 알고리즘의 모델 구성은 기본적으로 TD-MPC 모델 구성이랑 같다. 제안하는 알고리즘의 모델 구성은 식(18) - 식(22)와 같다. 식 (19) - 식(21)에서 $\ddot{a}_t$은 식 (14)에서 정의된 이중 정책을 활용한 TD-MPC를 통해 선택된 환경에 적용되는 행동이다.

$$Representation:\ z_t = h_\theta(s_t) \quad (18)$$

$$Latent\ Dynamics:\ z_{t+1} = d_\theta(z_t, \ddot{a}_t) \quad (19)$$

$$Reward:\ \hat{r}_t = R_\theta(z_t, \ddot{a}_t) \quad (20)$$

$$Action\ Value:\ \hat{q}_t = Q_\theta(z_t, \ddot{a}_t) \quad (21)$$

$$Policy:\ \hat{a}_t = \pi_\theta(z_t) \quad (22)$$

제안하는 알고리즘의 모델 구성은 TD-MPC와 기본적으로 유사하나, $\ddot{a}_t$은 식 (14)를 통한 이중 정책을 통하여 산출되는 것을 사용하는 것이 상이하다. 정책 모델을 제외한 모델의 업데이트는 식 (23)으로 이루어진다. 정책 모델은 DDPG 알고리즘을 활용 모델 업데이트를 한다.

3.4 제안하는 알고리즘 의사코드

그림 3는 본 논문에서 제안하는 알고리즘의 의사코드를 보여준다. 라인 3 - 라인 6은 제안하는 알고리즘의 행동 선택 비율을 조정하는 이중 정책 활용을 의미한다. 라인 7 - 라인 9는 내적 보상을 통해 탐험-활용 균형을 맞추는 것을 의미한다. 라인 10의 의미는 버퍼에 경험을 저장하여 훈련 시 임의의 트레젝토리를 활용하고자 하는 것이다. 라인 14 – 라인 22은 제안하는 알고리즘의 모델 업데이트를 의미하고 있다.

Algorithm 1 Proposed Algorithm

Require:
θ: randomly initialized network parameters
$\beta, \mathcal{B}$: intrinsic reward coefficient and buffer
ϵ: epsilon computed by linear scheduler
$\mathcal{T}, \mathcal{R}$: transition probability distribution
Π_θ: model-based policy
π_θ: model-free policy

1: **while** not tired **do**
2: **for** step t = 0...**T do**
3: **if** $random(0,1) < \epsilon$ **then**
4: $\ddot{a}_t \sim \Pi_\theta(\cdot|h_\theta(s_t))$
5: **else**
6: $\ddot{a}_t \sim argmax_{a_t \in (\Pi_\theta(\cdot|h_\theta(s_t)), \pi_\theta(\cdot|h_\theta(s_t)))} Q_\theta(s_t, a_t)$
7: $(s_{t+1}, r_t^{ext}) \sim \mathcal{T}(\cdot|s_t, \ddot{a}_t), \mathcal{R}(\cdot|s_t, \ddot{a}_t)$
8: $r_t^{int} = \mathcal{L}(h_\theta(s_{t+1}), d_\theta(h_\theta(s_t), \ddot{a}_t))$
9: $r_t = r_t^{ext} + \frac{\beta}{2} * r_t^{int}$
10: $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, \ddot{a}_t, r_t, s_{t+1})$
11: **for** num updates per episode **do**
12: Model Update

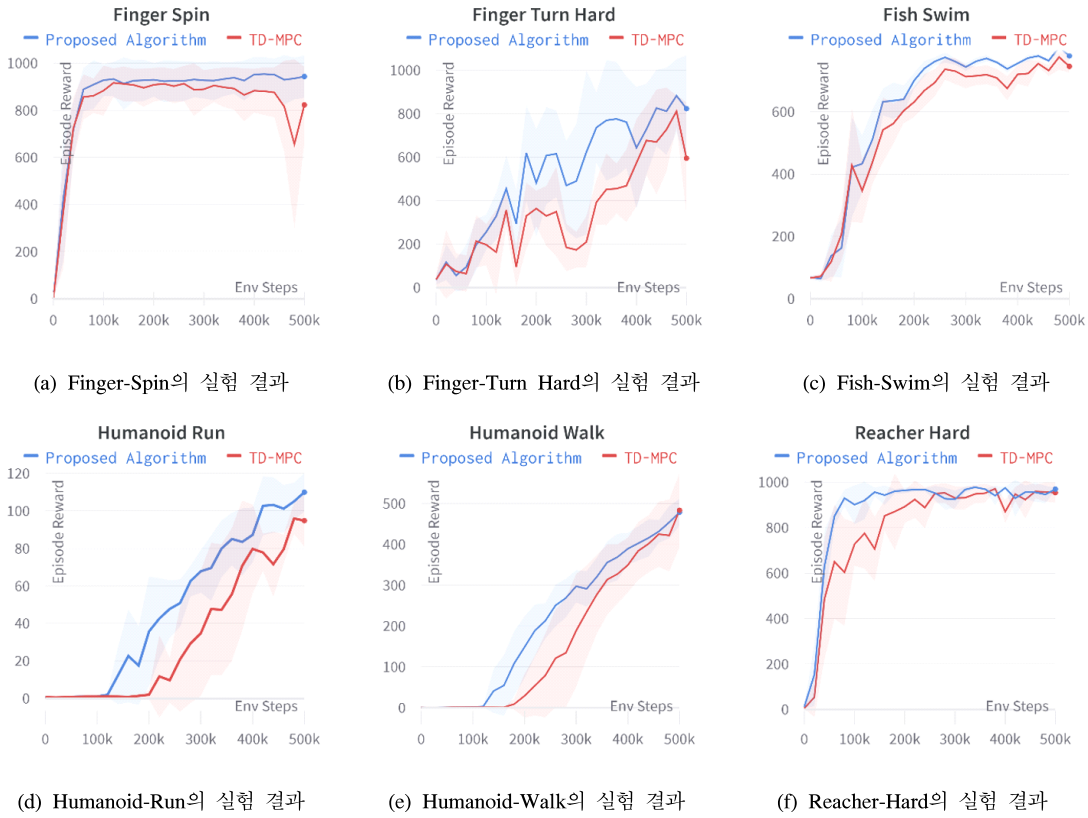
그림 3. 제안하는 알고리즘 의사코드
Fig. 3. Pseudo-code of proposed algorithm

Ⅳ. 실 험

DeepMind Control Suite는 강화학습 에이전트의 성능 벤치마크 역할을 하기 위한 표준화된 구조와 해석 가능한 보상이 포함된 일련의 연속 제어 작업이다 [2]. 본 논문에서는 DeepMind Control Suite의 여러 환경에서 TD-MPC와 제안하는 알고리즘의 성능을 비교한다. 다음 표 2는 각 환경의 상태 공간 및 행동 공간을 보여준다.

그림 4는 제안하는 알고리즘과 TD-MPC의 비교실험 결과 그래프를 보여준다. 총 6개의 환경에서 실험을 진행하였고, x축은 환경과 상호작용한 횟수를 y축은 에피소드 누적 보상이다. 환경마다 5번씩 실험을 하여 평균을 실선으로 표준편차를 범위로 나타내었다. 실험을 진행한 모든 환경에서 제안하는 알고리즘이 기존 TD-MPC보다 학습 속도가 빠르고, 성능도 더 높다는 것을 확인할 수 있다. 또한, 학습이 완료했을 때, 제안하는 알고리즘의 표준편차가 더 작은 것을 확인할 수 있는데, 이는 학습의 안정성도 제안하는 알고리즘이 더 높다는 것을 의미한다.

$$L(\theta) = c_1\|R_\theta(z_i, \ddot{a}_i) - (reward_{total})\|_2^2 + c_2\|Q_\theta(z_i, \ddot{a}_i) - ((reward_{total}) + \gamma Q_{\theta^-}(z_{i+1}, \pi_\theta(z_{i+1})))\|_2^2 + c_3\|d_\theta(z_i, \ddot{a}_i) - h_{\theta^-}(z_{i+1})\|_2^2 \quad (23)$$



(a) Finger-Spin의 실험 결과 (b) Finger-Turn Hard의 실험 결과 (c) Fish-Swim의 실험 결과

(d) Humanoid-Run의 실험 결과 (e) Humanoid-Walk의 실험 결과 (f) Reacher-Hard의 실험 결과

그림 4. 제안하는 알고리즘과 TD-MPC 비교실험 결과 그래프
Fig. 4. The proposed algorithm and TD-MPC comparison test result graph

표 2. DeepMind Control Suite 실험 환경의 관찰 공간 및 행동 공간
Table 2. Observation shape and Action space of DeepMind Control Suite experiment environment

Environment	Observation Shape	Action Space
Finger Spin	Box(-inf, inf, (9,), float32)	Box(-1.0, 1.0, (2,), float32)
Finger Turn Hard	Box(-inf, inf, (12,), float32)	Box(-1.0, 1.0, (2,), float32)
Fish Swim	Box(-inf, inf, (24,), float32)	Box(-1.0, 1.0, (5,), float32)
Humanoid Run	Box(-inf, inf, (67,), float32)	Box(-1.0, 1.0, (21,), float32)
Humanoid Walk	Box(-inf, inf, (67,), float32)	Box(-1.0, 1.0, (21,), float32)
Reacher Hard	Box(-inf, inf, (6,), float32)	Box(-1.0, 1.0, (2,), float32)

V. 결 론

TD-MPC는 모델 프리 정책과 모델 기반 정책을 가지고 있음에도, 모델 기반 정책의 행동만 환경에 적용한다. 성격이 전혀 다른 2개의 정책 중 하나의 정책만 사용하는 것은 더 좋은 행동 선택의 기회를 낭비하는 것이다.

본 논문에서는 TD-MPC가 가지고 있는 모델 프리 정책과 모델 기반 정책 중 더 좋은 정책의 행동을 환경에 적용하는 이중 정책 TD-MPC를 제안한다. 이때, 학습 초반에는 모델 기반 정책을 활용하는 것이 모델 프리 정책의 학습 속도에도 좋은 영향을 주는 것을 실험으로 증명하여, 모델 기반 정책의 선택 비율을 선형적으로 조정한다. 또한, 이중 정책 TD-MPC가 활용에 집중하여 탐험-활용 균형이 깨지는 것을 방지하기 위해 이중 정책 TD-MPC와 호기심 기반 탐험 방법을 결합한 새로운 알고리즘을 제안한다.

제안하는 알고리즘과 TD-MPC는 DeepMind Control Suite의 여러 환경에서 비교실험을 진행하였

다. 대부분 환경에서 기존 TD-MPC보다 제안하는 알고리즘의 성능이 뛰어난 것을 확인할 수 있다. 제안하는 알고리즘의 성능 향상은 기존 TD-MPC와 비교하여 추가적인 모델이나 계산복잡도가 필요 없다.

향후 본 논문에서 적용한 ICM을 이외에 여러 가지 탐험 기법을 TD-MPC에 적용하여 비교실험을 통해 더 효율적인 탐험 기법을 TD-MPC에 적용할 예정이다. 또한, 해당 알고리즘을 그대로 적용하는 것뿐만이 아닌, 모델 복잡도를 올리지 않는 선에서 탐험을 강화할 수 있는 TD-MPC에 특화된 탐험 기법을 연구할 예정이다.

References

[1] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *J. Intell. & Robotic Syst.*, vol. 86, no. 2, pp. 153-173, 2017. (https://doi.org/10.1007/s10846-017-0468-y)

[2] Y. Tassa, et al., "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018. (https://doi.org/10.48550/arXiv.1801.00690)

[3] M. A. Graule, et al., "SoMoGym: A toolkit for developing and evaluating controllers and reinforcement learning algorithms for soft robots," *IEEE Robotics and Automat. Lett.*, vol. 7, no. 2, pp. 4071-4078, 2022. (https//doi.org/10.1109/LRA.2022.3149580)

[4] C. Park, C. Jeong, J.-H. Yoo, and C. M. Kang, "Efficient reinforcement learning method for dynamic system control," *Trans. KIEE*, vol. 71, no. 9, pp. 1293-1301, 2022. (https://doi.org/10.5370/KIEE.2022.71.9.1293)

[5] J. Bhatia, et al., "Evolution gym: A large-scale benchmark for evolving soft robots," *Advances in NIPS*, vol. 34, pp. 2201-2214, 2021. (https://doi.org/10.48550/arXiv.2201.09863)

[6] N. Hansen, et al., "Temporal difference learning for model predictive control," *39th Int. Conf. Mach. Learn.*, vol. 162, pp. 8387-8406, 2022. (https://doi.org/10.48550/arXiv.2203.04955)

[7] Y. Tassa, et al., "Synthesis and stabilization of complex behaviors through online trajectory optimization," *2012 IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pp. 4906-4913, 2012. (https://doi.org/10.1109/IROS.2012.6386025)

[8] T. P. Lillicrap, et al., "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015. (https://doi.org/10.48550/arXiv.1509.02971)

[9] M. L. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, 2014. (https://doi.org/10.1002/9780470316887)

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT Press, 2018. (https://doi.org/10.1016/S1364-6613(99)01331-5)

[11] T. M. Moerland, et al., "Model-based reinforcement learning: A survey," *arXiv preprint arXiv:2006.16712*, 2020. (https://doi.org/10.48550/arXiv.2006.16712)

[12] L. Kaiser, et al., "Model based reinforcement learning for atari," *Int. Conf. Learn. Representations*, 2019. (https://doi.org/10.48550/arXiv.1903.00374)

[13] M. Janner, et al., "When to trust your model: Model-based policy optimization," *Advances in NIPS*, vol. 32, 2019. (https://doi.org/10.48550/arXiv.1906.08253)

[14] A. Rajeswaran, et al., "A game theoretic framework for model based reinforcement learning," *Int. Conf. Mach. Learn.*, PMLR, pp. 7953-7963, 2020. (https://doi.org/10.48550/arXiv.2004.07804)

[15] Y. Fan and Y. Ming, "Model-based reinforcement learning for continuous control with posterior sampling," *Int. Conf. Mach. Learn.*, PMLR, pp. 3078-3087, 2021. (https://doi.org/10.48550/arXiv.2012.09613)

[16] J. Schrittwieser, et al., "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604-609, 2020. (https://doi.org/10.1038/s41586-020-03051-4)

[17] W. Ye, et al., "Mastering atari games with limited data," *Advances in NIPS*, vol. 34, 2021. (https://doi.org/10.48550/arXiv.2111.00210)

[18] C. B. Browne, et al., “A survey of monte carlo tree search methods,” *IEEE Trans. Computat. Intell. and AI in games*, vol. 4, no. 1, pp. 1-43, 2012. (https://doi.org/10.1109/TCIAIG.2012.2186810)

[19] D. Pathak, et al., “Curiosity-driven exploration by self-supervised prediction,” *Int. Conf. Mach. Learn.*, PMLR, 2017. (https://doi.org/10.48550/arXiv.1705.05363)

[20] I. M. de Abril and R. Kanai, “Curiosity-driven reinforcement learning with homeostatic regulation,” *2018 IJCNN IEEE*, 2018. (https://doi.org/10.1109/IJCNN.2018.8489075)

[21] J. Li, et al., “Random curiosity-driven exploration in deep reinforcement learning,” *Neurocomputing*, vol. 418, pp. 139-147, 2020. (https://doi.org/10.1016/j.neucom.2020.08.024)

[22] G. Williams, et al., “Model predictive path integral control using covariance variable importance sampling,” *ArXiv, abs/1509.01149*, 2015. (https://doi.org/10.48550/arXiv.1509.01149)

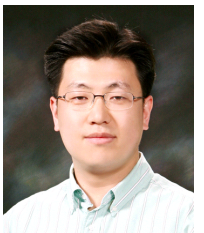
지 창 훈 (Chang-Hun Ji)

2020년 8월 : 한국기술교육대학교 메카트로닉스공학과 학사
2023년 2월 : 한국기술교육대학교 컴퓨터공학과 석사
2023년 3월~현재 : 한국기술교육대학교 컴퓨터공학과 박사과정
<관심분야> 사물인터넷, 정밀 제어, 강화학습

김 주 봉 (Ju-Bong Kim)

2017년 2월 : 한국기술교육대학교 컴퓨터공학과 학사
2019년 2월 : 한국기술교육대학교 컴퓨터공학과 석사
2019년 3월~현재 : 한국기술교육대학교 컴퓨터공학과 박사과정
<관심분야> 정밀 제어, 딥러닝, 멀티 에이전트 강화학습

한 연 희 (Youn-Hee Han)



1996년 2월 : 고려대학교 수학과 학사
1998년 2월 : 고려대학교 컴퓨터공학과 석사
2002년 2월 : 고려대학교 컴퓨터공학과 박사
2002년 3월~2006년 2월 : 삼성종합기술원 전문연구원
2013년 9월~2014년 8월 : SUNY at Albany, Department of Computer Science 방문교수
2006년~현재 : 한국기술교육대학교 컴퓨터공학부 교수
<관심분야> 사물인터넷, 5G/6G, 딥러닝, 강화학습, 조합최적화
[ORCID:0000-0002-5835-7972]